

基于自主运算的自适应存储区域网络系统*

付长冬⁺, 舒继武, 郑纬民, 沈美明

(清华大学 计算机科学与技术系 高性能计算技术研究所,北京 100084)

A Self-Adaptive Storage Area Network System Based on Autonomic Computing

FU Chang-Dong⁺, SHU Ji-Wu, ZHENG Wei-Min, SHEN Mei-Ming

(Institute of High Performance Computing Technologies, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-10-62783505, E-mail: fcd01@mails.tsinghua.edu.cn, <http://www.tsinghua.edu.cn>

Received 2003-06-11; Accepted 2004-02-03

Fu CD, Shu JW, Zheng WM, Shen MM. A self-adaptive storage area network system based on autonomic computing. *Journal of Software*, 2004,15(7):1056~1063.

<http://www.jos.org.cn/1000-9825/15/1056.htm>

Abstract: Storage management is one of the most important problems for the current network storage system. The kernel of solving storage management problem is to self-adapt storage system to all changes of external environment and internal events, and to carry out self-regulation and self-management. The key to storage manageability is self-adaptation. A new method and architecture for implementing storage management is introduced in this paper that is to realize self-adaptation SAN storage management system based on Autonomic Computing theory. On the basis of Tsinghua Mass storage network system project (TH-MSNS), we have produced a common framework of the SAN (storage area network) storage management system, and implemented an adaptive-level Autonomic SAN storage system. Test results show that the new SAN storage system has a better performance than the old system.

Key words: self-adaptive; storage management system; automatic computing; SAN (storage area network); intelligent network disk

摘要: 存储管理是目前网络存储系统亟需解决的重要问题之一,解决存储管理问题的核心是自动适应外部环境变化,实现自我调整和自我管理,其中自适应是管理的关键.提出了一个解决存储管理的新方法和体系结构,基于自主运算理论实现自适应的 SAN(storage area network)存储管理系统.在 Tsinghua Mass Storage Network System(TH-MSNS)项目的基础上,提出了一个基于自主运算的 SAN 存储管理系统的通用结构框架,实现了一个自适应级别的自主 FC-SAN 存储系统.初步的实验结果说明,新的 SAN 存储系统具有较好的性能.

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA111110 (国家高技术研究发展计划(863))

作者简介: 付长冬(1971—),男,江苏连云港人,博士生,主要研究领域为网络存储,嵌入式 OS;舒继武(1969—),男,博士,副教授,主要研究领域为网络存储,并行计算;郑纬民(1946—),男,教授,博士生导师,主要研究领域为并行计算,体系结构,网络存储,网络计算;沈美明(1938—),女,教授,博士生导师,主要研究领域为并行与分布处理.

关键词: 自适应;存储管理系统;自主运算;SAN;智能网络 DISK

中图法分类号: TP393 文献标识码: A

目前的 SAN(storage area network)存储系统存在许多问题,存储管理是 SAN 存储系统亟需解决的重要问题之一.因为 SAN 存储系统的复杂性,使得存储管理越来越复杂,不断地需要系统管理员去发现问题,并手工干预和管理,因此无法保证数据存储的高可靠性、高可用性,而且管理费用巨大.系统复杂性的主要因素包括:系统的规模在不断变化之间,动态地增加和删除存储设备、服务器、驱动器等;应用系统多种多样,如电子商务、数据库系统、分布式文件系统,不同应用产生不同特征的工作负载和存取模式.系统中可能包括多个厂商的设备和系统,具有较强的异构性;系统升级、重新启动、部件失败等造成系统重构;不同用户的共享安全控制等^[1~4].解决存储管理问题的核心是存储系统自动适应外部环境变化,实现自我调整和自我管理.目前还没有一个较好的体系和办法来解决存储管理问题.

WIND 项目^[5]通过在网络 DISK 中增加一些信息收集和处理模块,使网络 DISK 能够适应工作负载和其他情况的变化,实现可管理的存储设备.CMU 大学的 NASD 项目^[6,7]介绍了共享存储情况下的适应性的管理功能.共享存储模型^[8]详细描述了各种可能的共享存储的模型结构,分析了数据安全和数据共享的管理需求.PETAL 项目^[9]为基于集群的商业化 PC 实现了一个虚拟 DISK,这些 PC 能够共享存取虚拟化的 DISK.它实现了一些运行时的自适应功能,例如能够根据读请求进行动态的队列调整.HP 公司的 AUTORAID 项目^[10]是 RAID 子系统自我调整的一个例子,它实现 2 层的存储体系,根据数据的存取频率在不同速率的设备中存放数据,不同速率的设备负责存取频率不同的数据.TSS 项目^[11]介绍了存储管理系统的策略框架问题,并给出了一些简单的具体实现方法.上述工作只是根据项目的需要做了一些自适应的工作,但无法通用,也没有完整而清晰的方法和体系结构.

最近,IBM 研究中心提出了计算系统领域的一个新的理论:自主计算^[12,13].自主运算是在人工干预最少的前提下实现计算系统自动管理的一种方法.它能够使计算系统自我运行,不断地自我调整适应各种环境,并根据工作负载情况预先分配各种资源以便提高系统处理能力.我们的想法是,将自主运算理论引入到 SAN 存储管理领域,提出一个新的、完整的体系结构和方法,解决网络存储系统的可管理性问题.

Tsinghua Mass Storage Network System(TH-MSNS)项目实现了一个全部采用商业化的硬件完成的海量 SAN 存储器,软件系统包括 FCP 目标器驱动程序、SCSI 目标模拟器、I/O 子系统等^[14].在 TH-MSNS 项目的基础上实现了一个基于自主运算的自适应的 SAN 存储管理系统 SA-SSMS(self-adaptive SAN storage management system).一方面,将 SAN 存储系统的存储设备、服务器和光纤交换机实现为较高层的自主运算组件;另一方面,实现最高层的存储管理软件自主运算组件.最高层的存储管理软件将存储设备等自主运算组件作为可控制的复合资源,组成 SA-SSMS 的智能控制环.存储管理软件自主运算组件提供用户制定的策略,用来决定如何控制各种资源,而存储设备等自主运算组件根据自身的环境变化向上层组件提供所需要的资源信息,并且接受上层组件发送来的用户策略和决定,动态地配置、优化和调整自己以适应环境的变化.

SA-SMSS 提出了一个基于自主运算的 SAN 存储管理系统的结构框架,并按照自主运算可适应级的要求实现了智能网络 DISK、可管理的服务器、FC 交换机的智能代理等低层自主运算组件,同时还实现了高层的自主运算组件:存储管理软件,它们组成了一个完整的自主运算系统,实现了 SAN 存储系统的自我配置、自我规划、自我优化、自我恢复等功能.

1 基于自主运算的存储管理模型与方法

一个 SAN 存储系统的管理对象,如存储设备实现为自主运算组件,需要在管理对象内部实现至少一个自主运算的智能控制环.每个控制环包括 4 个模块:资源、测量、决定、控制,它们之间存在顺时针的控制接口关系.任何一个控制环的资源可以是系统低层的资源单元,如文件、数据,也可以是高层的复合资源单元,如核心驱动程序、OS、数据库、服务器等.每个资源存在与控制环中其他部分的标准接口:感应器和反应器.感应器提供

“GET”操作,而反应器提供“SET”操作.

基于自主运算的自适应的 SAN 存储管理系统的实现方法就是将 SAN 存储系统的管理对象,如交换机、存储设备、服务器等实现为自主运算组件,作为 SAN 存储管理软件自主运算组件内部智能控制环上的高层复合资源,这些资源一方面通过感应器 GET 接口向存储管理软件提供各种信息,另一方面通过反应器 SET 接口接收上层的参数、动作和策略.存储管理软件控制环上的决定模块根据用户制定的策略和通过测量部件得到的各种信息来作出有效控制各种资源、适合环境变化的最佳决定.控制环中的控制模块将根据决定将对应的参数、行动和策略反馈给下层资源的反应器.

一个基于自主运算的自适应的 SAN 存储管理系统的模型如图 1 所示(其中的①,②,③分别表示 measurement component,decision component,control component).

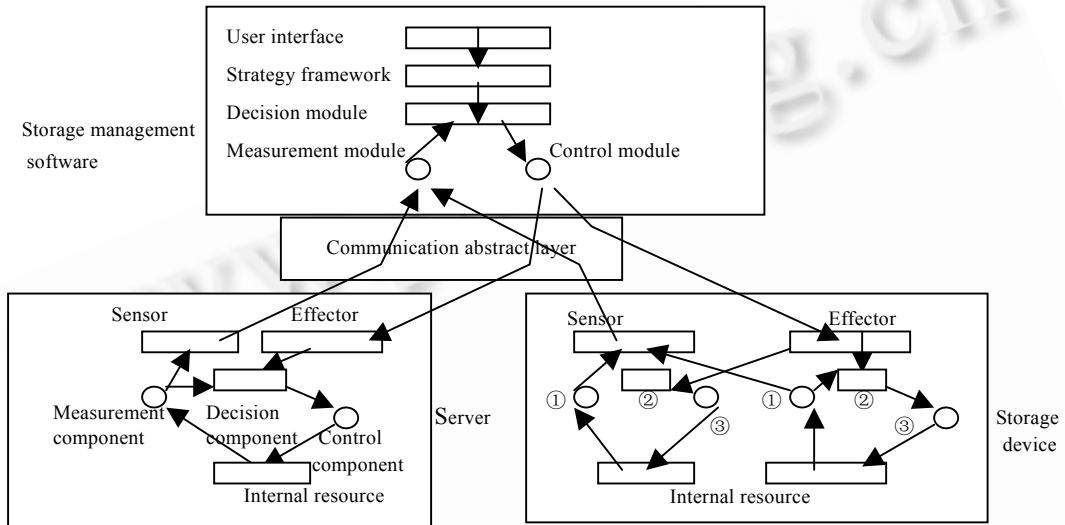


Fig.1 The model of SAN storage management system based on autonomic computing

图 1 基于自主运算的 SAN 存储管理系统模型

考虑到模型的通用性,SAN 存储管理系统的不同组件之间的通信细节内容由通信抽象层表示,它是一个点对多点的 C/S 关系的承载层软件,但它不是自主运算组件.SAN 存储管理软件是最高层的自主运算组件,负责制定和提供各种运行策略.它需要提供用户接口给用户,由用户根据最佳经验制定资源策略或其他策略,因此用户不再需要关心具体的管理动作,而仅仅需要制定高层的管理策略.策略框架可以包含用户制定的任意策略.

图 1 的 SAN 存储管理系统模型仅仅给出了服务器、存储设备等下层自主运算组件,实际应用中一个自主运算组件可以存在多个智能控制环,控制任意多个资源,如存储设备;一个自主运算组件可以同时提供多个不同的资源给同一个高层组件或不同的高层组件.

2 自适应的存储管理系统

2.1 体系结构

基于自主运算的 SAN 存储管理系统的模型和方法,在 TH-MSNS 项目的基础上实现了一个自适应的 SAN 存储管理系统 SA-SSMS.它包括 4 个高层的自主运算组件:

- SAN 存储管理软件,是最高层的自主运算组件,负责提供各种用户策略,并根据用户策略和各种资源的信息制定各种问题的解决方案.
- 智能网络 DISK,是一个可动态调整性能、优化数据放置并可自我配置的大容量的 FC-DISK.它与海量 SAN 存储器相比有很大的区别,一方面实现了智能控制环,另一方面,也是更重要的,为了适应外部环境的变化,增加了一些资源自主适应模块.

- 可管理的服务器,实现了一个应用层的代理软件:存储资源管理器,将服务器内的存储资源包括 SCSI 子系统、FCP 驱动程序封装成一个自主运算组件,负责自主管理和优化各服务器的存储资源和 I/O 请求等。

- FC 交换机的智能代理,它无法在封闭的交换机上实现自主运算组件,是 FC 交换机 FABRIC OS 的代理软件,通过标准的 SWAPI 接口函数实现一个管理交换机的自主运算组件,运行在 SAN 存储管理软件的主机上。

如图 2 所示,SAN 存储管理软件是最高层的自主运算组件,它管理的对象包括 SAN 存储设备、FC 交换机、前端的服务器,而这些自主运算组件间的通信抽象层采用 IP 协议的 SOCKET 接口实现。服务器上的存储资源管理器可以在 OS 核心实现,也可以在用户层实现。我们采用的是在用户层实现,图中只是为了清楚起见,将它和 SCSI 子系统等并列。FC 交换机智能代理软件相当于一个在 FC 交换机的 FABRIC OS 的外部实现一个自主运算组件,它通过 API 接口和 FABRIC OS 通信,FABRIC OS 作为 FC 交换机智能代理的内部资源。工作负载管理器和智能磁盘管理器是智能网络 DISK 内的两个子自主运算组件,因此智能网络 DISK(SAN 存储设备)提供两个复合资源给 SAN 存储管理软件。

自适应的 SAN 存储管理软件的内部结构(基于光纤通道)如图 2 所示。

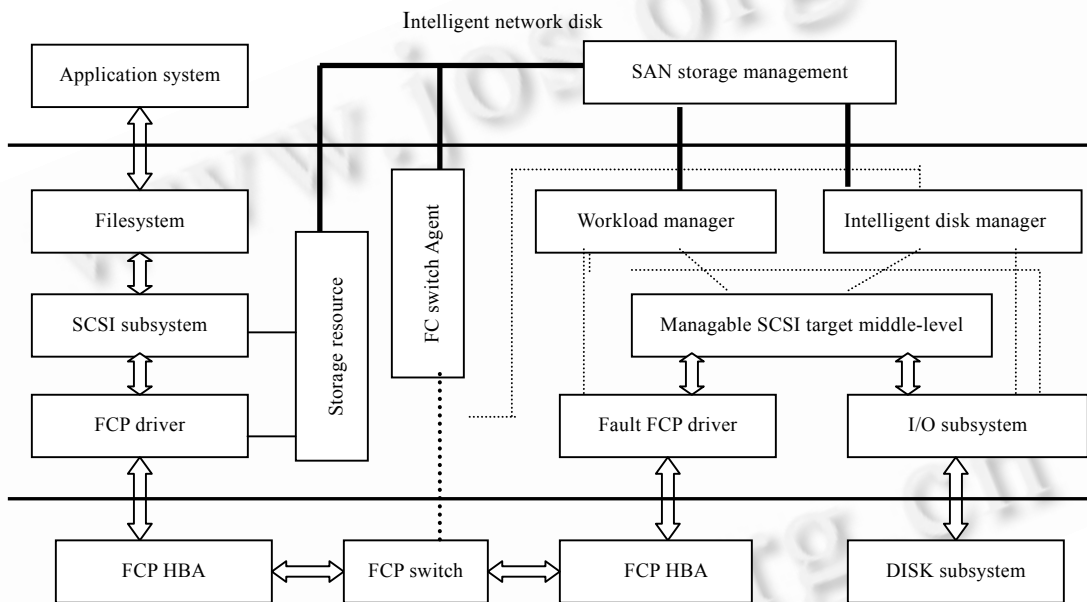


Fig.2 Software architecture of SAN storage management based on autonomic computing

图 2 基于自主运算的 SAN 存储管理软件结构

图 2 中双向箭头表示数据流程,而粗黑线表示存储管理软件的控制环线,其他表示内部的自主运算组件控制环线。

2.2 智能网络DISK

智能网络 DISK(IND)是一个大容量的支持多个 RAID 级别的 FC-DISK,硬件系统采用商业化的硬件部件实现。它的基本软件模块包括目标器模式的 FCP 驱动程序、SCSI 目标模拟器、I/O 子系统等。

- 目标器模式的 FCP 驱动程序,是 FC HBA 的目标器模式的 FCP 驱动程序,它与 FC HBA 的目标器模式的固件交互,接受 FCP IU(信息单元)并转化成 SCSI 命令给 SCSI 命令处理程序,并将 SCSI 命令转化为 FCP IU 在 SAN 网络上传输。

- SCSI 目标模拟器(STML),负责 SCSI 命令和消息的排队、调度处理,并且负责光纤通道 LUN(逻辑单元)和 I/O 子系统内部的磁盘设备之间的映射。

- I/O 子系统,存在不同的实现方式,根据各种 SCSI 命令进行具体的磁盘 I/O 处理。

为了实现自主运算化的智能网络 DISK,一方面需要将基本模块实现为自主运算组件,而且需要增加适应外

部环境变化和内部事件的资源自主适应模块(组件).

- 工作负载管理器,根据用户工作负载和存取模式进行资源调整 and 性能优化.
- 智能 DISK 管理器,根据磁盘的空间分布和存取情况进行存储空间的规划和调整,以满足在线管理、用户存取等需求.
- 动态事件管理器,对于各种突然发生的事件,如一个 RAID 中的一个磁盘失败,根据用户的策略作出自动处理,基本不需要用户干涉.

智能网络 DISK 的体系结构如图 3 所示.

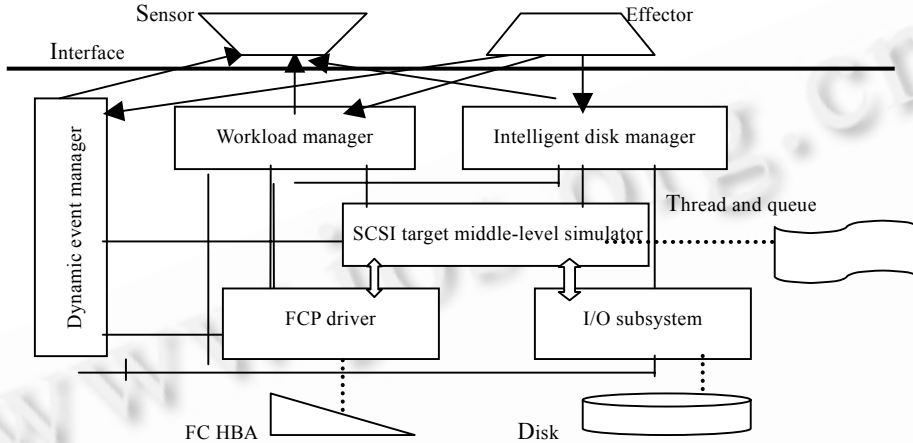


Fig.3 Internal software architecture of INFCD

图 3 INFCD 的内部软件体系结构

如图 3 所示,智能网络 DISK 通过感应器向 SAN 存储管理软件提供各种信息测量接口,而通过反应器接受各种参数、动作和策略.整个智能网络 DISK 向上层自主运算组件提供 3 个复合资源:事件、工作负载、存储空间.动态事件管理器、工作负载管理器、智能 DISK 管理器内部控制环的资源是 SCSI 目标模拟器、FCP 驱动程序和 I/O 子系统组件.SCSI 目标模拟器提供线程、SCSI 消息队列和命令队列资源.FCP 驱动程序提供 FC 链路和 IOCB 队列等资源.I/O 子系统提供存储空间资源.

SCSI 目标模拟器是智能网络 FC-DISK 中最核心的一个组件,它负责如何调度 SCSI 命令的执行.如图 4 所示,给出可管理的 SCSI 目标模拟器(SCSI target-mode middle-level module,简称 STML)组件的内部结构.在自主运算体系结构中,SCSI 命令队列、SCSI 消息队列和处理线程是 STML 控制环中的资源,需要进行有效的控制和管理.如图 4 所示.

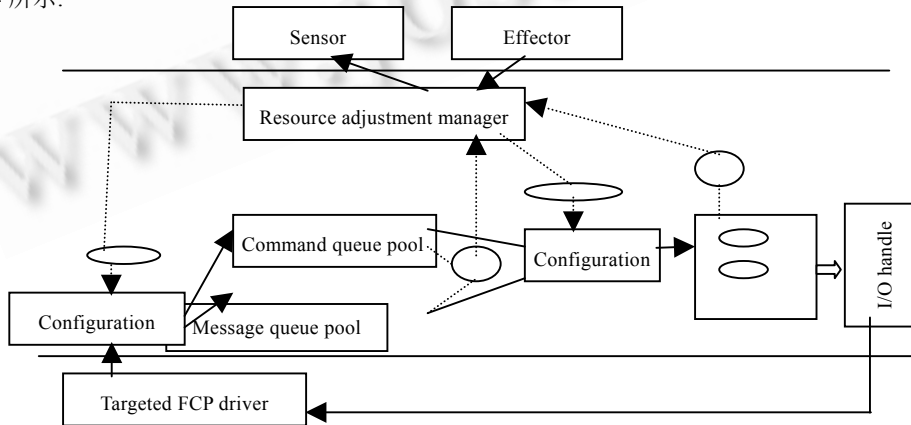


Fig.4 The architecture of STML autonomous computing component

图 4 STML 自主运算组件的体系结构

图 4 中的圆圈表示测量操作,而椭圆表示控制操作,控制操作设置具体的配置.实线表示数据流控制.

STML 自主运算组件的资源调节器(决定模块)负责从 3 个资源,即命令队列池、消息队列池、线程池中得到各种资源信息,如目前运行的进程数、第 X 个命令队列上的 outstanding SCSI 命令数等等.通过对这些信息的分析,调节 FCP 目标器驱动如何发送 SCSI 命令或消息到 SCSI 消息队列和 SCSI 命令队列,调节线程从哪个队列中取消息和命令进行处理,调节运行线程的数目等等.而 STML 的资源调节器进行调节的主要依据是工作负载管理器通过反应器设置的工作负载策略和资源策略.同时,STML 将在具体负载的资源情况下通过传感器将各种信息传输给工作负载管理器以及其他高层组件.STML 实现了两个智能控制环,分别控制消息和命令队列、线程资源.

2.3 可管理的SAN存储管理软件

自主运算的存储管理软件与以往的存储管理软件最大的区别在于,它不进行具体的管理操作而主要制定用户策略.它的核心是如何制定策略和如何作出正确的决定.SAN 存储管理软件自主运算组件的实现需要重点解决 3 个问题:(1) 与下层自主运算组件的传感器和反应器的接口和规范;(2) 策略框架的实现方法和每个策略项的结构;(3) 动态的决定模块.

SAN 存储管理软件管理的是不同厂商的设备、软件或系统,作为上层控制环中的资源而需要提供的感应器和反应器接口必须统一规范.感应器接口至少包括 5 类操作接口:资源、性能统计、配置参数、事件、策略;反应器的接口与普通的存储管理软件操作接口存在很大的区别.反应器不仅仅是设置参数,更主要的是设置用户策略.具体描述如下:

- 参数:值,配置下层组件的参数.
- 行动:具体的、定义好的一个动作,如不让用户 X 读 DISK N .
- 策略:一个条件和具体的行动.它告诉下层组件做什么?具体如何做,由下层组件决定.每个策略包括一个条件和多个动作.

因为为用户配置的策略往往不是最好的策略,因此需要不断地调整,以得到最佳策略.用户可以不断地从最佳策略中得到经验,从而形成默认的策略配置.一个自动优化、配置和恢复的动态策略循环模型如图 5 所示.

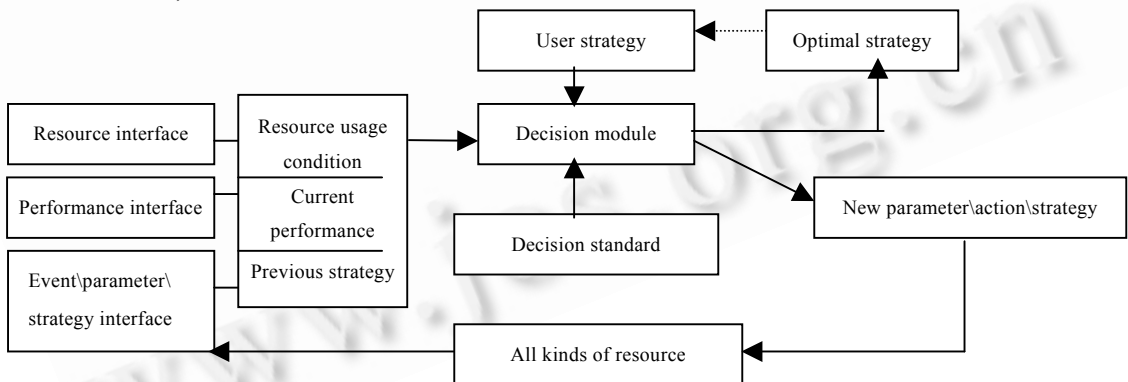


Fig.5 Loop model of dynamic decision

图 5 动态决策循环模型

如图 5 所示,各种判断原则来自于对顾问或专家经验的收集,它是用户制定的策略的判断依据,决定模块会通过各资源的感应器得到的信息,与判断原则进行比较,判断用户目前制定的策略是否能够满足用户需求.例如一个判断原则为:当用户数在 50 以上,而每个用户的平均响应时间为 1MS 以下,则该策略就是一个好策略.用户输入的是初始策略,最后得到的是最佳策略.决定模块根据用户目前的策略项,决定对应的自主运算组件模块(资源)应该采取的行为(参数、行动或策略).

存储管理软件中的策略项可以是动态增加的,也可以是静态设置的,并且可以按照模块、类型或者一个统一模式的方式进行组织.一个较好的方法是动态地按照模块方式组织策略框架,策略管理比较简单,而且策略清

晰,但不好的方面是一个策略是针对一个自主运算组件的,具有贪婪算法的缺点,无法得到系统的最佳性能比.同时,为了方便统一配置、统一 RESET 等操作,所有的策略按照参数、行动和策略的类型实现 3 个双向链表.

SAN 存储管理软件通过 3 个行为方式管理和控制下层的自主运算组件:配置参数、采取一个特定的动作,或者传递一个特定的策略给下层组件作为控制策略.它的策略项可以包括各种各样的策略,如用户共享控制策略、事件处理策略、性能策略、资源策略等等.通过这些策略实现自主运算系统的自我配置、自我管理、自我恢复、自我保护等核心属性.

3 系统原型

我们实现了基于自主运算的 SAN 存储管理系统的原型——SA-SSMS,其硬件平台由 5 个 IA-64 服务器、一个 16 端口的 SILMWORM3800 交换机以及由自己开发实现的智能网络 DISK 组成.

服务器的操作系统采用 LINUX 2.4.18 核心.服务器上的资源管理器自主运算组件在用户空间实现,它直接利用用户空间的 HBA API 库函数和核心层的 SCSI 子系统和 FC HBA 驱动程序通信实现.可以实现的功能包括:完善的信息获取、I/O 路径管理以及事件管理,并修改了驱动程序,增加了 SCSI 命令传输管理、设备失败处理等功能,以便在用户端实现负载控制,增加的功能采用 PROC 用户接口提供给资源管理器.为了模拟 FC 交换机的自主运算功能,我们还利用 4 个 FC HBA 在 LINUX 操作系统下实现了一个模拟的光纤通道交换机.FC 交换机的智能代理(自主运算组件)运行在 SAN 存储管理软件的主机上,该智能代理将交换机 FABRIC OS 作为控制环中的资源,采用 SWAPI 和 FABRIC OS 进行通信,可以实现 LUN 屏蔽、用户分区等粗粒度的安全控制策略,并可以实现 FC 链路的控制管理.我们还利用一个模拟的 FC 交换机实现了 FC 传输流根据工作负载变化的自适应功能,FC 交换机是实现性能自我优化最好的自主运算组件.

SAN 存储软件与下层自主运算组件之间采用通用信息模型 CIM(common information model)规范作为框架标准,可以很好地解决兼容性问题.基于 CIM 的 SAN 存储管理软件框架在 SAN 存储管理软件和下层的自主运算组件之间增加了 CIM 对象管理器和对象提供者,主要目的是兼容多个厂商的不同产品,从而降低系统的管理和构建复杂性.SAN 存储管理软件采用 JAVA 实现,而其他自主运算组件基本上采用 C 语言实现.自适应的 SAN 存储系统的通信抽象层基于 TCP/IP 协议实现.

图 6 给出了 5 个服务器共享一个由 5 个 SCSI 磁盘组成的 RAID0 磁盘的性能情况,以及相同条件下基于自主运算的自适应的 FC-SAN 存储系统的吞吐量性能.

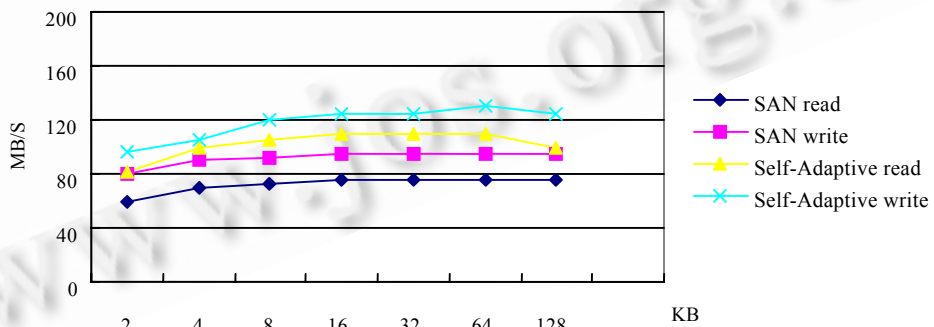


Fig.6 Throughput of shared disk

图 6 共享磁盘的吞吐量性能

如图 6 所示,给出了原有 SAN 存储系统的读写性能以及基于自主运算的自适应的 FC-SAN 存储系统的读写性能.基于自主运算的 FC-SAN 存储系统的吞吐量大约比原来系统增加了 25%左右,而且在 16KB 左右,基本达到了系统性能的最大值.而随着数量块的增大,系统性能呈略微下降趋势,这主要是因为传输时间的增加而导致的.

4 结束语

存储管理是目前网络存储系统需要解决的重点问题之一,自适应是存储管理的关键.自适应的管理系统可以根据外部环境的动态变化和内部事件,自动调整,自我管理,从而能够提供较好的性能,减少人工干预,降低系统的复杂性,实现可管理性.本文将自主运算理论引入到存储管理系统中,提出了一个实现可适应的 SAN 存储管理系统的新方法和体系结构,构建了一个具有普遍意义的基于自主运算的存储管理系统的结构框架.基于自主运算的可适应的 SAN 存储管理系统,具有自我恢复、自我配置、自我优化、自我保护等自主运算属性,将所有服务器、存储设备和交换机作为不同的资源统一纳入到一个智能控制环中,根据用户制定的策略动态地调整优化整个 SAN 存储系统,较好地解决了存储管理的问题.下一步的核心工作是建立一个较好的策略框架和资源管理模型,实现可适应性较高的 SAN 存储系统.

References:

- [1] Talagala N, Patterson D. An analysis of error behaviour in a large storage system. Technical Report, UCB/CSD-99-1042, Berkeley: Computer Science Division, University of California, 1999.
- [2] Meter RV. Observing the effects of multi-zone Disks. In: Kohl J, ed. Proc. of the 1997 USENIX Conf. Anaheim: USENIX Press, 1997.
- [3] Arpaci-Dusseau RH. Performance availability for networks of workstations [Ph.D. Thesis]. Berkeley: University of California, 1999.
- [4] Riedel E, Kallahalla M, Swaminathan R. A framework for evaluating storage system security. In: Long D, ed. Proc. of the 1st Conf. on File and Storage Technologies. Monterey: USENIX Press, 2002. 15~30.
- [5] Arpaci-Dusseau AC, Arpaci-Dusseau RH, Bent J, Forney B, Muthukrishnan S, Popovici F, Zaki O. Manageable storage via adaptation in WiND. In: Raje R, ed. Proc. of the IEEE Int'l Symp. on Cluster Computing and the Grid. Brisbane: IEEE Computer Society Press, 2001. 169~177.
- [6] Gibson GA, Nagle DF, Amiri K, Butler J, Chang FW, Gobioff H, Hardin C, Riedel E, Rochberg D, Zelenka J. A cost-effective, high-bandwidth storage architecture. ACM SIGPLAN Notices, 1998,33(11):92~103.
- [7] Gibson GA, Nagle DF, Amiri K, Chang FW, Gobioff H, Riedel E, Rochberg D, Zelenka J. Filesystems for network-attached secure disks. Technical Report, CMU-CS-97-118, Carnegie Mellon University, 1997.
- [8] Wilkes J, Rickard W, Gibson G, Anderson D, Black D. Shared storage model-a framework for describing storage architecture. Technical Council Proposal Document, draft-june5, 2001.
- [9] Lee EK, Thekkath CA. Petal: Distributed virtual disks. In: Dally B, ed. Proc. of the 7th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems. Cambridge: ACM Press, 1996. 84~92.
- [10] Golding WR, Staelin C, Sullivan T. The HP AutoRAID hierarchical storage system. ACM Trans. on Computer Systems, 1996,14(1):108~136.
- [11] Pallipadi V. Design, implementation and policy framework for a Linux based temperature sensitive storage. In: Andregg BC, ed. Proc. of the 5th Annual Linux Showcase and Conf. Oakland: USENIX Press, 2001. 107~118.
- [12] Draper C. Autonomic computing architecture: A blueprint for Managing complex computing environments. 2002. <http://www.research.ibm.com/autonomic/>
- [13] IBM Corp. Autonomic computing concepts. 2002. http://www-306.ibm.com/autonomic/pdfs/AC_Concepts.pdf
- [14] Palekar AA, Russell RD. Design and implementation of a SCSI target for storage area networks. Technical Report, TR 01-01, Durham: Computer Science Department, University of New Hampshire, 2001.